

AMENDMENTS TO THE CLAIMS:

1. (Currently Amended) A method for recovery of processing in a host in ~~of a component~~ of a distributed network accounting system, ~~the method comprises:~~ the host including at least one node and at least one component, the method comprising:

context check-pointing state of processing in the at least one component to operably permit automatic recovery of the at least one component to the component's most recent processing context checkpoint using operating system facilities; and

executing the at least one node in the host under the control of a node manager that persistently maintains the current state of nodes in the host, the current state comprising process status and a list of usage data that has been successfully processed.

~~executing operating system facilities to provide automatic recovery of the system components to the component's most recent processing context.~~

2. (Currently Amended) The method of claim 1 wherein context check pointing state of processing in the at least one component further comprises:

maintaining the component's processing context as in-memory object and as a disk-based file.

3. (Currently Amended) The method of claim [[1]] 7 wherein executing operating system facilities ~~facility~~ further comprises:

re-starting the component from its last check-pointed processing context during a subsequent recovery of the component in the system.

4. (Currently Amended) The method of claim 1 wherein the node manager is a component within the host, and the host further comprises a data manager component. ~~components are nodes where changes in the processing context of the component are characterized as generally single/atomic transactions or other transactions that are easily check-pointed.~~

5. (Currently Amended) The method of claim [[4]] 1 further comprising:

executing the at least one component as an “Immortal” process under management of the operating system facilities.

6. (Currently Amended) The method of claim [[4]] 1 further comprising:

executing the at least one component as an “Immortal” process thereby operably causing the operating system facilities to automatically re-start the component from its most recent checkpoint state in case of failure.

7. (Currently Amended) The method of claim 1 further comprising:

executing operating system facilities to provide the automatic recovery of the component to the component's most recent processing context. ~~executing each host in the system under the control of a node manager that persists to disk the current state of the host, where the current state includes the process status of which components are executing and a list of usage data that has been successfully processed.~~

8. (Original) The method of claim 1 wherein in the event of a graceful or non-graceful shutdown of a component or the system, the system state is preserved, and is used to restore the system back to its last known state.

9. (Currently Amended) The method of claim 8 wherein the at least one component stores copies of records received for backup and restore purposes, enabling the records to be re-processed in the event of a downstream system failure.

10. (Currently Amended) A computer program product residing on a computer readable medium for recovering a state of processing in a host in component of a distributed network accounting system, ~~comprises instructions for causing a computer to:~~ the host including at least one component and at least one node, the program product comprising instructions for causing a computer to:

context check-point a state of processing in the at least one component to operably permit automatic recovery of the at least one component to the component's most recent processing context checkpoint using operating system facilities; and

execute the at least one node in the host under the control of a node manager that persistently maintains the current state of nodes in the host, the current state comprising process status and a list of usage data that has been successfully processed.

~~execute an operating system facility to provide automatic recovery of the system component to the component's most recent processing context.~~

11. (Currently Amended) The computer program product of claim 10 wherein instructions to context check point the state of processing in the at least one component further comprise instructions to:

maintain the component's processing context as in-memory object and as a disk-based file.

12. (Currently Amended) The computer program product of claim 10 wherein instructions to context check point the state of processing in the at least one component further comprise instructions to:

re-start the component from its last check-pointed processing context during a subsequent recovery of the component in the system.

13. (Currently Amended) The computer program product of claim 10 wherein instructions to execute further comprise instructions to:

execute the component as an “Immortal” process under management of the operating system facilities to operably cause the operating system facilities to automatically re-start the component from its most recent checkpoint state in case of failure.

14. (Currently Amended) A distributed network accounting system, comprising:

a plurality of host computers that hosts at least one network accounting process and a computer program product residing on a computer readable medium for providing fault tolerance to a data processing domain for the [[a]] network accounting process, each of the host computers including at least one component and at least one node, the program product comprising ~~comprises~~ instructions for causing at least one of the host computers ~~computer~~ to:

context check-point a state of processing in the at least one component to operably permit automatic recovery of the at least one component to the component's most recent processing context checkpoint using operating system facilities; and

execute the at least one node in the host under the control of a node manager that persistently maintains the current state of nodes in the host, the current state comprising process status and a list of usage data that has been successfully processed.

~~context check-point a state of processing in the a data processing domain to permit automatic recovery of the data processing domain to the data processing domain's most recent processing context checkpoint; and~~

~~execute an operating system facility to provide the automatic recovery of the data processing domain to the data processing domain's most recent processing context.~~

15. (Original) The distributed network accounting system of claim 14 wherein the data processing domain is a run-time node manager, a run-time data manager or an administrative configuration manager.

16. (Original) The distributed network accounting system of claim 14 wherein the data processing domain further comprises:

a recovery manager that executes the computer program product to recover a state of the data processing domain.

17. (Original) The distributed network accounting system of claim 16 wherein the at least one network accounting process is a data collector process that produces network accounting records, or an aggregation process that aggregates network accounting records, or an enhancement process that enhances attributes of network accounting records, or an output interface process that produces records for use by an application.

18. (New) The method of claim 1 wherein the at least one component is the node manager and the host computer further comprises a data manager component, the method further comprising:

context check-pointing state of processing in the data manager.

19. (New) The method of claim 18 wherein data manager component comprises a local data manager and a remote data manager, and the method further comprises:

context check-pointing state of processing in the local data manager; and

context check-pointing state of processing in the remote data manager.

20. (New) The method of claim 1 wherein the host further comprises a data manager component, the method further comprising:

maintaining the at least one node and the data manager in memory as objects by the node manager.